Physique Cuso Python 2013 (PCP13) – EPFL
Introduction

Valentin Haenel

Freelance Consultant and Software Developer
http://haenel.co

Feb 2013

Version: 2013-02-pcp13.1 https://github.com/pcp13/intro

Outline

1. Introduction
2. Materials and Communication
3. Survey
4. Schedule
5. Software Installation

The Premise

> *Scientists spend an increasing amount of time building and using software. However, most scientists are never taught how to do this efficiently. As a result, many are unaware of tools and practices that would allow them to write more reliable and maintainable code with less effort.*

- D. A. Aruliah et. al. *Best Practices for Scientific Computing.* Dec 2012

- See also: http://software-carpentry.org/

About the Speaker

- Freelance consultant and software developer
- Experience as scientific software developer
- Author of a book about version control with Git
- Tutorial chair for EuroScipy 2011 and 2012
- Frequent contributor to open source software
- Background in computer science and computational neuroscience

Goals

- The goal of this course, is not for you to learn everything there is to know about scientific computing with Python.
- For me, it is important to give you the map of the scientific Python ecosystem:
    - What are the popular tools and libraries?
    - What is their basic premise, how do they work?
    - Which tool for what purpose?
    - Where can I find more information?
    - What are the best practices?
- So, please don't be disappointed, if at first things don't make sense, it will take you a while to grasp the big picture.

Python Versions

- Currently there are two main lines of python 2.x and the 3.x line
  - The 2.x versions in use are 2.5, 2.6 and 2.7
  - The 3.x version in use is 3.3
- From 2.x to 3.x many long standing warts and issues have been resolved
- As a result, 3.x is not backwards compatible
- This is causing some strain on the community and the library authors
- For scientific code, many of the main libraries have been ported
- **BUT** many clusters and super-computers still only support 2.x
- We will be using 2.7

Outline

1. Introduction
2. Materials and Communication
3. Survey
4. Schedule
5. Software Installation

Material

- Material mainly based on the scipy-lecture-notes
    - http://scipy-lectures.github.com/
- But drawing from other sources, e.g. Python Summerschool as and when needed
    - https://python.g-node.org/wiki/
- Was written by different people, so please excuse the potential differences in style

Licencing

- Almost all material under a creative commons licence http://creativecommons.org/licenses/by-sa/3.0/
- ⇒ Allows me (and yourself):
  - **to Share** — to copy, distribute and transmit the work
  - **to Remix** — to adapt the work
  - to make commercial use of the work
- ⇒ Requires you to:
  - Retain the attribution of the original author
  - Share are publish only under the same or similar license to this one.
- Sources (`*.tex *.wiki *.rst`) provided
- Will indicate which slides are not under this licence

Github

- Github organization: `pcp13`
  - https://github.com/organizations/pcp13
  - Make a Gitub account, send me your username by email to join
- All materials (PDF Slides, schedule, news items etc..) will be in the https://github.com/pcp13/orga repository.
  - Follow the RSS feed of commits to stay up-to-date
  - https://github.com/pcp13/orga/commits/master.atom
  - The repository is free to clone, you don't strictly need a Github account to participate
- Additional repositories for available sources
  - May be forks of other repositories
  - May be private repositories
- At the end of the course these repositories will be frozen
  - ⇒ You may return to exact state, should you need to look something up again

Github

Github and Accounts

- I have received a free educational account from Github for this course
  - ⇒ 50 free private repositories
  - Each of you is entitled to one of these
  - You are invited to use it to store your solutions and collaborate with your classmates
  - Send your Github username to the email address listed in the next slide
- Github offers free micro plans for students
  - ⇒ 5 free private repositories
  - Use for your class projects, thesis, or research towards a degree

Communication, E-Mail and X

- You can send email to `pcp13@haenel.co`
- Use the prefix `[pcp13]` in the subject
- If you want to use twitter (I don't), I suggest the hashtag: `#pcp13`
- Make you own Facebook (I don't use that either) group, if you want to

Obtaining Credits

- There will be a one hour (maybe longer) exam on Friday afternoon
- Pencil and paper exercises
- Don't panic... It will be doable!

Outline

1. Introduction
2. Materials and Communication
3. Survey
4. Schedule
5. Software Installation

Survey - Which Operating System?

- Linux
- Mac OSX
- Windows
- Other

Survey - Which Programming Languages?

- Matlab
- C/C++
- Java
- Python

- Other

Survey - Version Control

- Who knows what this is?
- Who has used it before?
  - Subversion
  - Git
  - Other

Schedule

- Let's visit the orga repository and have a look at the preliminary schedule
- Depending on how well we make progress, we may have to reschedule
  - Check the orga repository for updates
- General Idea:
  - Start at 8:00 (or 9:00)
  - Do a lecture for 1.5 - 2 hours
  - Take a coffee break, 0.5 hours
  - Do some exercises for 1.5 - 2 hours
  - Have lunch for 1 - 1.5 hours
  - Do a lecture for 1.5 - 2 hours
  - Take a coffee break, 0.5 hours
  - Do some exercises for 1.5 - 2 hours
- If you don't need the credits, you are welcome to join only those lectures that you are interested in

Outline

1. Introduction
2. Materials and Communication
3. Survey
4. Schedule
5. Software Installation

Software Installation

We will now proceed to install some software on your machines

- Python
- Git

Python

- We have the choice of several pre-built Python distributions
- Anaconda CE from Continuum Analytics
  - Free community edition
  - Pay for premium features
- Enthought Python Distribution (EPD) from Enthought
  - Free for academic use
  - Requires registration and login to download
- Python(x,y) and Winpython
  - Two free distributions
- If you are using Linux or Mac OSX use distribution packages or Homebrew/Fink/...

Git

- Git doesn't come in too many flavours
- For Linux, definitely use the packages available from your distribution
  - e.g. `sudo aptitude install git`
- For Mac OSX and Windows use the binary installers

Links

- Anaconda
  - https://store.continuum.io/cshop/anaconda
  - http://anaconda.zetatech.org/ ← alternative
- EPD
  - http://www.enthought.com/products/epd.php
- Python(x,y)
  - http://code.google.com/p/pythonxy/
- Winpython
  - http://code.google.com/p/winpython/
- Git
  - http://git-scm.com/downloads

... and don't forget to make a Github account ;-)